

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-247423

(43)Date of publication of application : 19.09.1997

(51)Int.Cl.

H04N 1/387  
H03M 7/30  
H04N 1/41

(21)Application number : 08-046294

(71)Applicant : CANON INC

(22)Date of filing : 04.03.1996

(72)Inventor : NAKAYAMA TADAYOSHI

## (54) IMAGE PROCESSOR AND ITS METHOD

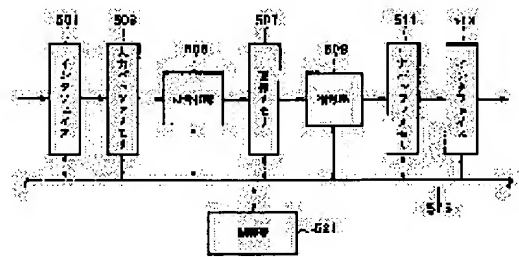
### (57)Abstract:

PROBLEM TO BE SOLVED: To reduce the storage capacity of a memory storing image data in the image processor and its method applying image rotation processing to the image data.

SOLUTION: A coding section 505 applies Joint Photographic Expert Group(JPEG) coding to inputted image data and adds a restart marker to the coded image data every time the image data are coded by a prescribed amount and stores the coded image data to an image memory 507.

A decoding section 509 reads the coded data out of the image memory 507 and decodes the image data based on the restart mark in a strip form in the subscanning direction and stores the decoded image data to an output buffer memory 511. A

control section 521 reads image data stored in the output buffer memory 511 sequentially in the subscanning direction and sends the data to an interface 513.



BEST AVAILABLE COPY

## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than  
the examiner's decision of rejection or  
application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's  
decision of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

**\* NOTICES \***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

**DETAILED DESCRIPTION**

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the image processing system which processes the inputted image data, and its approach, concerning an image processing system and its approach.

[0002]

[Description of the Prior Art] It becomes the procedure of transmitting the image data which once stored the image data which received all the image data for one image, and received the image which the image data which received expresses 90 degrees or when carrying out rotation 270 degrees (clockwise rotation) (henceforth "image rotation processing") and re-transmitting in image memorandum \*\* etc., and was stored in the image memory perpendicularly with read-out.

[0003] Drawing 1 is a flow chart explaining image rotation processing.

[0004] In this drawing, the image data which received image data at step S101, and received at step S103 is stored in an image memory. And S103 is repeated from step S101 until it judges whether all the image data of one image was received and finishes receiving at step 104.

[0005] If all the image data of one image is received and stored, the read-out starting position of the image data according to angle of rotation of an image will be decided (S104), it will scan from the starting position to the perpendicular direction of an image, and image data will be transmitted for image data with read-out (S105) (S106). And S107 is repeated from step S105 until it judges whether all image data was transmitted and finishes transmitting at step S107.

[0006] In this case, in the case of a 4096x4096-pixel 24-bit full color image, the memory capacity of a needed image memory turns into memory capacity of no less than 48 M bytes depending on the maximum image size. Then, it is possible to reduce the storage capacity of an image memory using the JPEG (Joint Photographic Experts Group) method which is the international standards of a multiple-value digital still picture compression method. This compression method carries out orthogonal transformation of the picture signal using DCT (Discrete Cosine Transform), reduces information by removing the signal component of a part which is not visually conspicuous with the signal after conversion, and after that, in order to compress information further, it performs Huffman coding etc. Huffman coding assigns the variable-length code according to the occurrence frequency of a sign, in order to reduce the amount of signs. Therefore, in order to decode the encoded Huffman code, it is necessary to decode

sequentially from the head of a sign.

[0007] Drawing 2 is a flow chart which shows the example which applied JPEG coding to the image rotation processing shown in drawing 1. In addition, the same sign is given to the step which performs the same processing as drawing 1, and the detail explanation is omitted to it.

[0008] In drawing 2, JPEG compression of the image data which received at step S201 is carried out, and the image data compressed at step S202 is stored in an image memory.

[0009] If all the image data of one image is received, compressed and stored, the compression image data stored in the image memory is decoded (S203), and it stores in buffer memory (S204). And it judges whether decode of compression image data was completed at step S205, and S205 is repeated from step S203 until decode is completed.

[0010] When all the compression image data of one image is decoded and being stored in buffer memory, the same processing as S107 will be performed from step S104 shown in drawing 1, and the image data of the image which rotated will be transmitted.

[0011]

[Problem(s) to be Solved by the Invention] However, there are the following troubles in the technique mentioned above. That is, when rotating the image which the image data which received expresses and transmitting the image data of the image which rotated, there is a problem that the mass memory which can memorize all the image data that received is needed.

[0012] Even if it compresses a metaphor and the image data which received and memorizes to an image memory, when rotating and transmitting an image Since the image data which decoded the compressed image data, stored in buffer memory etc., and was stored in buffer memory is scanned perpendicularly and read-out and the procedure of transmitting are needed Although the memory capacity of an image memory is reducible, there is a problem that the memory capacity of the buffer memory for storing the decoded image data becomes large depending on image size.

[0013] This invention aims at reducing the storage capacity of the memory which stores image data in the image processing system which it is and performs image rotation processing to image data and its approach for solving an above-mentioned problem.

[0014]

[Means for Solving the Problem] This invention is equipped with the following configurations as a way stage which attains the aforementioned purpose.

[0015] This invention is characterized by providing the following in an image processing system. A coding means to add a predetermined control code to the code data whenever it carries out specified quantity coding of said image data, while carrying out sequential coding of the inputted image data in the main scanning direction A storage means to memorize the image data encoded by said coding means The decode means which decodes the code data read from said storage means based on said predetermined control code, and carries out the sequential output of the decoded image data in the direction of vertical scanning

[0016] Moreover, the coding means which carries out JPEG coding of the inputted image data and the first storage means which memorizes the image data encoded by said coding means, A decode means to decode code data in read-out and predetermined size from said first storage means, It is characterized by having the second storage means which memorizes the image data decoded by said decode means, and an output means to read

the image data memorized by said second storage means in predetermined order, and to output it.

[0017] This invention is characterized by providing the following in the image-processing approach. The coding step which carries out sequential coding of the inputted image data in the main scanning direction The addition step which adds a predetermined control code to the code data, and is stored in a storage means whenever it carries out specified quantity coding of said image data The decode step which decodes the code data read from said storage means based on said predetermined control code The output step which carries out the sequential output of the decoded image data in the direction of vertical scanning

[0018] Moreover, the coding step which carries out JPEG coding of the inputted image data, and stores the encoded image data in the first storage means, Code data is decoded in read-out and predetermined size from said first storage means, and it is characterized by having the decode step which stores the decoded image data in the second storage means, and the output step which reads the image data memorized by said second storage means in predetermined order, and outputs it.

[0019]

[Embodiment of the Invention] Hereafter, the image processing system of 1 operation gestalt concerning this invention is explained to a detail with reference to a drawing.

[0020] In each following operation gestalt, subject-copy image data are full color data of 8 bits each of RGB, and after they change this into a YUV color space, they shall carry out JPEG coding with the sampling rate of 4:1:1. In this case, the magnitude of 1MCU (the minimum coding unit) becomes 16x16 pixels.

[0021]

[The 1st operation gestalt] This operation gestalt reduces the storage capacity of buffer memory required to store the decoded image data by decoding compression image data per stripe of predetermined width of face. And in order to decode compression image data per stripe of predetermined width of face, with this operation gestalt, the restart marker who is the control code of JPEG is inserted in the image data which carried out JPEG coding (addition).

[0022] It is drawing for explaining in which location of the subject-copy image before compression drawing 3 inserts a restart marker. The subject-copy image data before compression shall be made into 4096x4096 pixels to level and a perpendicular direction, and shall insert a restart marker every horizontal (main scanning direction) 256 pixels.

Since 1MCU is 16x16 pixels as mentioned above, at least 2 bytes of restart marker will be inserted every 16MCU. Since JPEG coding is a variable-length-coding method, a restart marker's spacing in the information after coding (this is called bit stream) becomes in addition, less fixed. Furthermore, the band-like image field before the compression divided by the restart marker will be called a stripe. The number of the stripes in drawing 3 is set to  $4096 / 256 = 16$ .

[0023] Drawing 4 is the block diagram showing the example of a configuration of the image processing system of this operation gestalt.

[0024] The interface whose 501 receives input image data in this drawing, The input buffer memory which stores temporarily a part of image data which 503 received, The coding section which carries out JPEG coding of the image data by which 505 was stored in the input buffer memory 503, The image memory in which the image data by which

JPEG coding of 507 was carried out is stored by one image, The decode section which carries out the JPEG decode of the code data by which 509 was stored in the image memory 507, the output buffer memory which stores temporarily the image data by which 511 was decoded, and 513 are the interfaces for outputting the image data stored in the output buffer memory 511 to the exterior. 521 is a control section, for example, consists of CPU, a ROM, RAM, etc., and controls each above-mentioned block through a bus 515 according to the program beforehand stored in ROM.

[0025] Drawing 5 A is the flow chart which shows the example of a procedure of the coding processing in this operation gestalt, and is controlled and performed by the control section 521 mentioned above.

[0026] In drawing 5 A, the image data which received image data at step S401, and received at step S402 is stored in the input buffer memory 503. Since the image data for 16 lines is the need, it judges whether the image data for 16 lines was received, or all image data was received at step S403, and as mentioned above, if it is unfinished, it will return to step S401 performing JPEG coding.

[0027] When the image data for 16 lines or all image data are received, the image data which performed JPEG coding at step S404, and was encoded at step S405 is stored in an image memory 507. And it judges again whether reception of image data was completed at step S406, if it is termination, coding processing will be ended, otherwise, it progresses to step S407.

[0028] Then, it judges whether coding for 16MCU was completed, or coding of a block line (minimum Rhine required to perform coding processing) was completed at step S407, and if that is right, a restart marker will be inserted at step S408. And it judges again whether coding of a block line was completed at step S409, and if it is termination, since it is necessary to memorize a restart marker's location inserted after coding termination of a block line, after storing a restart marker's positional information in the pointer for every block line, it returns to step S401 at step S410. Moreover, if coding of a block line was not completed, it will return to step S404. In addition, a pointer is assigned to RAM of a control section 521 etc.

[0029] Drawing 5 B is the flow chart which shows the example of a procedure of the image rotation processing in this operation gestalt, and is controlled and performed by the control section 521 mentioned above.

[0030] The image data which decoded the code data stored in the image memory 507 at step S421, and was decoded at step S422 is stored in the output buffer memory 511. And S423 is repeated from step S421 until it detects a restart marker at step S423.

[0031] The positional information of the restart marker who detected the restart marker at step S424 when the image data for 1MCU width of face of detection, i.e., one stripe, was decoded is stored in a pointer, and the code data for a predetermined restart marker is skipped at step S425. With this operation gestalt, the code data for 15 restart markers will be skipped. That is, although the code data corresponding to the image data under the Rhine part equivalent to 1MCU of the same stripe will be read, it judges [ which skipped at step S426 ] whether there is any code data previously, if there is code data, it will return to step S421, and since it will mean that the decode for one stripe had ended if there is nothing, it progresses to step S427.

[0032] Then, image rotation processing mentioned above at step S427 in the image data for one stripe stored in the output buffer memory 511 is performed.

[0033] Next, the processing not more than step S421 which moved to processing of the following stripe at step S429, and described above whether the value (a restart marker's positional information) of a pointer and the positional information of the restart marker of the last block line recorded by coding processing would be in agreement at step S428 if processing was ended and it was not in agreement when in agreement [ judged and ] is repeated.

[0034] Thus, since according to this operation gestalt it decodes one stripe of code data at a time and image rotation processing is performed based on a restart marker, storage capacity required for buffer memory is reducible. For example, if width of face of a stripe is made into 256 bytes, it is 256x4096x3 bytes, i.e., 3 M bytes, and storage capacity of buffer memory can be made into 1/16 compared with the conventional method (48 M bytes).

[0035]

[The 2nd operation gestalt] Hereafter, the image processing system of the 2nd operation gestalt concerning this invention is explained. In addition, in the 2nd operation gestalt, about the same configuration as the 1st operation gestalt and abbreviation, the same sign is attached and the detail explanation is omitted.

[0036] In the 1st operation gestalt mentioned above, although the example which inserts a restart marker for every stripe was explained, as shown in drawing 6, a restart marker is inserted in the tail (or head) of a block line in this operation gestalt. Therefore, the processing at the time of coding is simplified a little compared with processing of the 1st operation gestalt.

[0037] Drawing 7 A is the flow chart which shows the example of a procedure of the coding processing in this operation gestalt, is controlled and performed by the control section 521 mentioned above, and explains only a different part from the procedure shown in drawing 5 A.

[0038] Although it judged whether coding for 16MCU was completed, or coding of a block line was completed at step S407 of drawing 5 A, it judges whether coding of a block line was completed in step S407a of drawing 7 A.

[0039] And after coding of a block line is completed, and inserting a restart marker at step S408, and there being no processing corresponding to step S409 of drawing 5 A, then storing a restart marker's positional information in the pointer for every block line at step S410, it returns to step S401.

[0040] Drawing 7 B is the flow chart which shows the example of a procedure of the image rotation processing in this operation gestalt, and is controlled and performed by the control section 521 mentioned above.

[0041] The code data for 1MCU of one block line is decoded at step S801. The positional information of code data [ finishing / decode ] is set to the pointer corresponding to a block line at step S802. The image data decoded at step S803 is stored in the output buffer memory 511. Each DC component is stored. Y, U, and V of the block decoded to the register of the lot corresponding to a block line at step S804 -- If judge and it is [ whether there is any next block line and ] at step S805, it will move to processing of the next block line at step S806, and processing of S805 will be repeated from step S801. In addition, a register is assigned to RAM of a control section 521 etc.

[0042] After processing of all the block lines equivalent to 1MCU width of face is completed, image rotation processing mentioned above at step S807 is performed, when

it is judged and detected whether the restart marker was detected at step S808, processing is ended, otherwise, it returns to step S801.

[0043] That is, in this operation gestalt, the image data of 1MCU width of face is processed as one stripe. and Y, U, and V which were stored in the register of the lot corresponding to a block line -- the code data read from the image memory 507 based on the pointer corresponding to a block line is decoded by each DC component. And in order to switch a block line for every decode processing of 1MCU and to switch a block line, it is not necessary to detect a restart marker, and a restart marker is used in order to detect whether all the code data were decoded.

[0044] In addition, although the cutting tool unit was sufficient since a restart marker's location was carrying out cutting tool alignment of the positional information of the restart marker who stores in a pointer in the 1st operation gestalt, the positional information stored in a pointer in the 2nd operation gestalt becomes bitwise finer than it. But the increasing information is only a triplet slightly.

[0045] Thus, since according to this operation gestalt it decodes per stripe of the width of face equivalent to 1MCU and image rotation processing is performed, storage capacity of the output buffer memory 511 can be made into 1/16 compared with the 1st operation gestalt.

[0046] Although \*\*\*\* explained the example which decodes per stripe of the width of face equivalent to 1MCU, and performs image rotation processing, it decodes per stripe of the width of face equivalent to a part for a MCU  $k$  ( $\geq 1$ ) individual, and may be made to perform image rotation processing. If it does in this way, the storage capacity of the output buffer memory 511 can make quick the part and process speed which decrease in the change of processing, although it increases  $k$  times compared with processing of the stripe unit of the width of face equivalent to 1MCU.

[0047]

[The 3rd operation gestalt] Hereafter, the image processing system of the 3rd operation gestalt concerning this invention is explained. In addition, in the 3rd operation gestalt, about the same configuration as the 1st operation gestalt and abbreviation, the same sign is attached and the detail explanation is omitted.

[0048] Next, the example which applies image-rotation-processing of the 1st or 2nd operation gestalt mentioned above to printers, such as an electrophotography method, is explained.

[0049] In case A4 recording paper is outputted, in order to make print speed quick in the printer of the electrophotography method corresponding to A3 recording paper generally, the printed output of the sense of the recording paper is carried out and carried out to a landscape in many cases. By the way, in almost all cases, the \*\*\*\*\* data sent to a printer from a host computer are premised on the sense of the recording paper being a portrait. That is, the sense of the premised detail paper in case data are transmitted to a printer from a host computer will differ from the sense of the detail paper which a printer actually outputs. Then, rotation processing of an image is needed.

[0050] The data transmitted to a printer from a host computer are divided roughly into the so-called PDL code and so-called image data which are represented by PostScript, PCL, etc. In case the former consists of a graphic command, a character code, etc. and bit map data are made, the so-called drawing processing is needed. Therefore, in case drawing processing is performed, the difference in the sense of the recording paper can be easily



absorbed depending on how to take system of coordinates. On the other hand, since the latter is already bit map data when it is transmitted to a printer from a host computer, image rotation processing is needed after transfer operation.

[0051] Drawing 8 is the block diagram showing the example of a configuration of the printer which JPEG-encodes, decodes a receipt, among those image data for two kinds of above-mentioned data from a host computer, and performs image rotation processing. In addition, the same sign is given to the configuration of drawing 4 and abbreviation identitas, and the detail explanation is omitted in it.

[0052] In this drawing, the drawing section which draws bit map data according to the PDL code to which 905 has been sent from the host computer, and 909 are the engine interfaces for outputting printer engine 911 HEBITTO map data.

[0053] The data received from the host computer through the interface 901 distinguish the PDL code and image data, and are stored in the input buffer memory 503. It is read to the coding section 505 for every data for 16 lines, and the image data stored in the input buffer memory 503 is stored in an image memory 507, after being compressed, as mentioned above. The compressed data stored in the image memory 507 is read by the decode section 509, after the decode and image rotation processing which were mentioned above using the output buffer memory 511 are performed, is read according to the printing speed of printer engine 911, and is sent to printer engine 911 through the engine interface 909.

[0054] On the other hand, the PDL code of the amount equivalent to the image for 1 page is accumulated in the input buffer memory 503, and in order to draw sequentially from what should be first outputted to printer engine 911, in advance of actual drawing processing, sorting of the case of the PDL code is beforehand carried out in data. In order to double the sense of the recording paper in that case, if required, system of coordinates will be rotated.

[0055] After reception of the PDL code equivalent to 1 page is completed, the PDL code is changed into bit map data by the drawing section 905, and it is stored in the output buffer memory 511. The bit map data stored in the output buffer memory 511 are read according to the printing speed of printer engine 911, and are sent to printer engine 911 through the engine interface 909.

[0056]

[Other operation gestalten] In addition, even if it applies this invention to the system which consists of two or more devices (for example, a host computer, an interface device, a reader, a printer, etc.), it may be applied to the equipments (for example, a copying machine, facsimile apparatus, etc.) which consist of one device.

[0057] Moreover, it cannot be overemphasized by the purpose of this invention supplying the storage which recorded the program code of the software which realizes the function of the operation gestalt mentioned above to a system or equipment, and carrying out read-out activation of the program code with which the computer (or CPU and MPU) of the system or equipment was stored in the storage that it is attained. In this case, the function of the operation gestalt which the program code itself read from the storage mentioned above will be realized, and the storage which memorized that program code will constitute this invention. As a storage for supplying a program code, a floppy disk, a hard disk, an optical disk, a magneto-optic disk, CD-ROM, CD-R, a magnetic tape, the memory card of a non-volatile, ROM, etc. can be used, for example.

[0058] Moreover, it cannot be overemphasized that it is contained also when the function of the operation gestalt which performed a part or all of processing that OS (operating system) which is working on a computer is actual, based on directions of the program code, and the function of the operation gestalt mentioned above by performing the program code which the computer read is not only realized, but was mentioned above by the processing is realized.

[0059] Furthermore, after the program code read from a storage is written in the memory with which the functional expansion unit connected to the functional expansion card inserted in the computer or a computer is equipped, it cannot be overemphasized that it is contained also when the function of the operation gestalt which performed a part or all of processing that CPU with which the functional expansion card and functional expansion unit are equipped based on directions of the program code is actual, and mentioned above by the processing is realized.

[0060]

[Effect of the Invention] As explained above, according to this invention, in the image processing system which performs image rotation processing to image data, and its approach, the storage capacity of the memory which stores image data is reducible.

---

[Translation done.]

(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-247423

(43) 公開日 平成9年(1997)9月19日

(51) Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N 1/387			H 0 4 N 1/387	
H 0 3 M 7/30		9382-5K	H 0 3 M 7/30	Z
H 0 4 N 1/41			H 0 4 N 1/41	B

審査請求 未請求 請求項の数10 O L (全 9 頁)

(21) 出願番号 特願平8-46294

(22) 出願日 平成8年(1996)3月4日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 中山 忠義

東京都大田区下丸子3丁目30番2号 キヤ  
ノン株式会社内

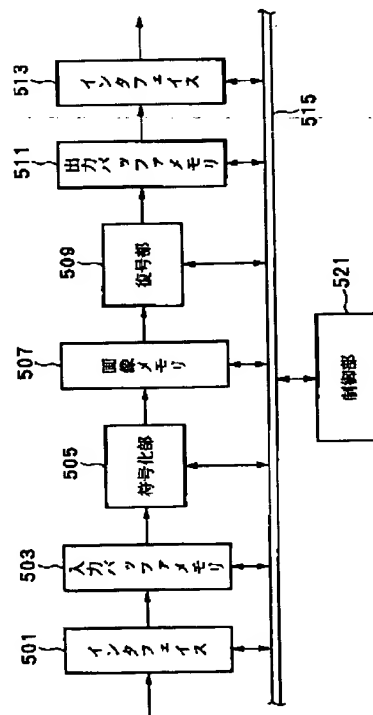
(74) 代理人 弁理士 大塚 康徳 (外1名)

## (54) 【発明の名称】 画像処理装置およびその方法

## (57) 【要約】

【課題】 画像データに画像回転処理を施すためには大容量のメモリが必要になる。

【解決手段】 符号化部505は、入力された画像データをJPEG符号化し、画像データを所定量符号化することによりリスタートマーカを付加し、符号化した画像データを画像メモリ507に格納する。復号部509は、画像メモリ507から符号データを読み出し、リスタートマークに基づき、画像データをその副走査方向に帯状に復号し、復号した画像データを出力バッファメモリ511へ格納する。制御部521は、出力バッファメモリ511に記憶された画像データを、その副走査方向に順次読出してインタフェイス513へ送る。



## 【特許請求の範囲】

【請求項1】 入力された画像データを、その主走査方向に順次符号化するとともに、前記画像データを所定量符号化するとともに、その符号データに所定の制御コードを付加する符号化手段と、

前記符号化手段により符号化された画像データを記憶する記憶手段と、

前記記憶手段から読出した符号データを前記所定の制御コードに基づいて復号し、復号した画像データを、その副走査方向に順次出力する復号手段とを有することを特徴とする画像処理装置。

【請求項2】 入力された画像データをJPEG符号化する符号化手段と、

前記符号化手段により符号化された画像データを記憶する第一の記憶手段と、

前記第一の記憶手段から符号データを読み出し、所定サイズ単位で復号する復号手段と、

前記復号手段により復号された画像データを記憶する第二の記憶手段と、

前記第二の記憶手段に記憶された画像データを所定の順に読み出して出力する出力手段とを有することを特徴とする画像処理装置。

【請求項3】 前記符号化手段は、前記画像データを所定量符号化するとともに、その符号データに所定の制御コードを付加することを特徴とする請求項2に記載された画像処理装置。

【請求項4】 前記復号手段は、前記所定の制御コードを検出すると、次のブロックラインの復号に移ることを特徴とする請求項3に記載された画像処理装置。

【請求項5】 前記復号手段は、画像データをその副走査方向に帯状に復号することを特徴とする請求項1または請求項4に記載された画像処理装置。

【請求項6】 前記出力手段は、前記帯状に復号された画像データを前記副走査方向に順に読み出すことを特徴とする請求項5に記載された画像処理装置。

【請求項7】 前記出力手段は、前記第二の記憶手段に記憶された画像データに画像回転処理を施すことを特徴とする請求項2に記載された画像処理装置。

【請求項8】 前記所定の制御コードは、JPEGの制御コードであるリスタートマーカであることを特徴とする請求項3に記載された画像処理装置。

【請求項9】 入力された画像データを、その主走査方向に順次符号化する符号化ステップと、

前記画像データを所定量符号化するとともに、その符号データに所定の制御コードを付加して記憶手段に格納する付加ステップと、

前記記憶手段から読出した符号データを前記所定の制御コードに基づいて復号する復号ステップと、

復号した画像データを、その副走査方向に順次出力する出力ステップとを有することを特徴とする画像処理方

法。

【請求項10】 入力された画像データをJPEG符号化し、符号化した画像データを第一の記憶手段に格納する符号化ステップと、

前記第一の記憶手段から符号データを読み出し、所定サイズ単位で復号し、復号した画像データを第二の記憶手段に格納する復号ステップと、

前記第二の記憶手段に記憶された画像データを所定の順に読み出して出力する出力ステップとを有することを特徴とする画像処理方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は画像処理装置およびその方法に関し、例えば、入力された画像データを処理する画像処理装置およびその方法に関するものである。

## 【0002】

【従来の技術】 受信した画像データの表示画像を90度あるいは270度（時計方向）回転して（以下「画像回転処理」という）、再転送する場合、一つの画像分の画像データをすべて受信し、受信した画像データを画像メモリなどに一旦格納し、画像メモリに格納された画像データを、垂直方向に読み出しながら転送するという手順になる。

【0003】 図1は画像回転処理を説明するフローチャートである。

【0004】 同図において、ステップS101で画像データを受信し、ステップS103で受信した画像データを画像メモリに格納する。そして、ステップS104で一つの画像の画像データをすべて受信したか否かを判定し、受信し終えるまでステップS101からS103を繰り返す。

【0005】 一つの画像の画像データをすべて受信し格納すると、画像の回転角度に応じた画像データの読み出し開始位置を決め（S104）、その開始位置から画像の垂直方向に走査して画像データを読み出し（S105）ながら、画像データを転送する（S106）。そして、ステップS107で画像データをすべて転送したか否かを判定し、転送し終えるまでステップS105からS107を繰り返す。

【0006】 この際に必要になる画像メモリの記憶容量は最大画像サイズに依存し、例えば、4096×4096画素の24ビットフルカラー画像の場合は48Mバイトもの記憶容量になる。そこで、多値デジタル静止画圧縮方式の国際標準であるJPEG (Joint Photographic Experts Group) 方式を用いて、画像メモリの記憶容量を削減することが考えられる。この圧縮方式は、DCT (Discrete Cosine Transform) を用いて画像信号を直交変換し、変換後の信号で視覚的に目立たない部分の信号成分を去除することで情報を減らし、その後、さらに情報を圧縮するためにハフマン符号化などを行う。ハフマン符号化は、符号量を減らすために、符号の発生頻度に応じた可変長のコードを割り当てるものである。そのため、符号化されたハフマ

ンコードを復号するには、符号の先頭からシーケンシャルに復号する必要がある。

【0007】図2は、図1に示した画像回転処理に、JPEG符号化を適用した例を示すフローチャートである。なお、図1と同様の処理を行うステップには、同一符号を付して、その詳細説明を省略する。

【0008】図2において、ステップS201で受信した画像データをJPEG圧縮し、ステップS202で圧縮した画像データを画像メモリに格納する。

【0009】一つの画像の画像データをすべて受信し圧縮し格納すると、画像メモリに格納された圧縮画像データを復号し(S203)、バッファメモリへ格納する(S204)。そして、ステップS205で圧縮画像データの復号が終了したか否かを判定し、復号が終了するまでステップS203からS205を繰り返す。

【0010】一つの画像の圧縮画像データをすべて復号してバッファメモリに格納すると、図1に示したステップS104からS107と同様の処理を行い、回転した画像の画像データを転送することになる。

【0011】

【発明が解決しようとする課題】しかし、上述した技術においては、次のような問題点がある。つまり、受信した画像データの表す画像を回転し、回転した画像の画像データを転送する場合、受信した画像データをすべて記憶することができる大容量のメモリが必要になるという問題がある。

【0012】例えば、受信した画像データを圧縮して画像メモリに記憶したとしても、画像を回転して転送する場合は、圧縮した画像データを復号してバッファメモリなどに格納し、バッファメモリに格納した画像データを垂直方向に走査して読出し、そして転送する、という手順が必要になるので、画像メモリの記憶容量は削減できるものの、復号した画像データを格納するためのバッファメモリの記憶容量が画像サイズに依存して大きくなるという問題がある。

【0013】本発明は、上述の問題を解決するためのものであり、画像データに画像回転処理を施す画像処理装置およびその方法において、画像データを格納するメモリの記憶容量を削減することを目的とする。

【0014】

【課題を解決するための手段】本発明は、前記の目的を達成する一手段として、以下の構成を備える。

【0015】本発明にかかる画像処理装置は、入力された画像データを、その主走査方向に順次符号化するとともに、前記画像データを所定量符号化するとともに、その符号データに所定の制御コードを付加する符号化手段と、前記符号化手段により符号化された画像データを記憶する記憶手段と、前記記憶手段から読出した符号データを前記所定の制御コードに基づいて復号し、復号した画像データを、その副走査方向に順次出力する復号手段

とを有することを特徴とする。

【0016】また、入力された画像データをJPEG符号化する符号化手段と、前記符号化手段により符号化された画像データを記憶する第一の記憶手段と、前記第一の記憶手段から符号データを読出し、所定サイズ単位で復号する復号手段と、前記復号手段により復号された画像データを記憶する第二の記憶手段と、前記第二の記憶手段に記憶された画像データを所定の順に読出して出力する出力手段とを有することを特徴とする。

【0017】本発明にかかる画像処理方法は、入力された画像データを、その主走査方向に順次符号化する符号化ステップと、前記画像データを所定量符号化するとともに、その符号データに所定の制御コードを付加して記憶手段に格納する付加ステップと、前記記憶手段から読出した符号データを前記所定の制御コードに基づいて復号する復号ステップと、復号した画像データを、その副走査方向に順次出力する出力ステップとを有することを特徴とする。

【0018】また、入力された画像データをJPEG符号化し、符号化した画像データを第一の記憶手段に格納する符号化ステップと、前記第一の記憶手段から符号データを読出し、所定サイズ単位で復号し、復号した画像データを第二の記憶手段に格納する復号ステップと、前記第二の記憶手段に記憶された画像データを所定の順に読出して出力する出力ステップとを有することを特徴とする。

【0019】

【発明の実施の形態】以下、本発明にかかる一実施形態の画像処理装置を図面を参照して詳細に説明する。

【0020】以下の各実施形態においては、原画像データはRGB各8ビットのフルカラーデータで、これをYUV色空間へ変換した後、4:1:1のサンプリングレートでJPEG符号化するものとする。この場合、1MCU（最小符号化単位）の大きさは16×16画素になる。

【0021】

【第1実施形態】本実施形態は、所定幅のストライプ単位で圧縮画像データを復号することにより、復号した画像データを格納するのに必要なバッファメモリの記憶容量を削減するものである。そして、所定幅のストライプ単位で圧縮画像データを復号するために、本実施形態では、JPEG符号化した画像データにJPEGの制御コードであるリスタートマーカを挿入（付加）する。

【0022】図3は圧縮前の原画像のどの位置にリスタートマーカを挿入するかを説明するための図である。圧縮前の原画像データは、水平・垂直方向に4096×4096画素とし、水平方向（主走査方向）の256画素ごとにリスタートマーカを挿入するものとする。上述したように、1MCUは16×16画素なので、16MCUごとに例えば少なくとも2バイトのリスタートマーカを挿入することになる。なお、JPEG符号化は可変長符号化方式であるため、符号

化後の情報（これをビットストリームという）におけるリスタートマーカの間隔は一定ではなくなる。さらに、リスタートマーカで区切られた圧縮前の帯状の画像領域をストライプと呼ぶことにする。図3におけるストライプの数は $4096/256=16$ になる。

【0023】図4は本実施形態の画像処理装置の構成例を示すブロック図である。

【0024】同図において、501は入力画像データを受信するインタフェイス、503は受信した画像データの一部を一時的に格納する入力バッファメモリ、505は入力バッファメモリ503に蓄えられた画像データをJPEG符号化する符号化部、507はJPEG符号化された画像データを一つの画像分蓄える画像メモリ、509は画像メモリ507に蓄えられた符号データをJPEG復号する復号部、511は復号された画像データを一時的に格納する出力バッファメモリ、513は出力バッファメモリ511に蓄えられた画像データを外部へ出力するためのインタフェイスである。521は制御部で、例えばCPU、ROM、RAMなどからなり、ROMに予め格納されたプログラムに従い、バス515を介して上記の各ブロックを制御する。

【0025】図5Aは本実施形態における符号化処理の手順例を示すフローチャートで、前述した制御部521により制御・実行されるものである。

【0026】図5Aにおいて、ステップS401で画像データを受信し、ステップS402で受信した画像データを入力バッファメモリ503へ格納する。前述したように、JPEG符号化を行うのに16ライン分の画像データが必要なので、ステップS403で、16ライン分の画像データを受信したか、または、画像データをすべて受信したかを判定し、未了であればステップS401へ戻る。

【0027】16ライン分の画像データ、または、画像データをすべて受信した場合は、ステップS404でJPEG符号化を行い、ステップS405で符号化された画像データを画像メモリ507へ格納する。そして、ステップS406で画像データの受信が終了したか否かを再度判定して、終了であれば符号化処理を終了し、そうでなければステップS407へ進む。

【0028】続いて、ステップS407で、16MCU分の符号化が終了したか、または、ブロックライン（符号化処理を行うのに必要な最小ライン）の符号化が終了したかを判定し、そうであれば、ステップS408でリスタートマーカを挿入する。そして、ステップS409でブロックラインの符号化が終了したか否かを再度判定して、終了であれば、ブロックラインの符号化終了後に挿入したリスタートマーカの位置を記憶しておく必要があるため、ステップS410で、各ブロックラインごとのポインタにリスタートマーカの位置情報を格納した後、ステップS401へ戻る。また、ブロックラインの符号化が終了したのでなければステップS404へ戻る。なお、ポインタは制御部521のRAMなどに割当てて。

【0029】図5Bは本実施形態における画像回転処理の手順例を示すフローチャートで、前述した制御部521により制御・実行されるものである。

【0030】ステップS421で画像メモリ507に格納された符号データを復号し、ステップS422で復号された画像データを出力バッファメモリ511に格納する。そして、ステップS423でリスタートマーカを検出するまで、ステップS421からS423を繰り返す。

【0031】リスタートマーカを検出、つまり1ストライプの1MCU幅分の画像データが復号されたら、ステップS424で検出したリスタートマーカの位置情報をポインタに格納し、ステップS425で所定リスタートマーカ分の符号データを読み飛ばす。本実施形態では、15個のリスタートマーカ分の符号データを読み飛ばすことになる。つまり、同じストライプの1MCUに相当するライン分下の画像データに対応する符号データを読み出すことになるが、ステップS426で読み飛ばした先に符号データがあるか否かを判定して、符号データがあればステップS421へ戻り、なければ1ストライプ分の復号が済んだことになるのでステップS427へ進む。

【0032】続いて、ステップS427で、出力バッファメモリ511に格納された1ストライプ分の画像データに前述した画像回転処理を施す。

【0033】次に、ステップS428で、ポインタの値（リスタートマーカの位置情報）と、符号化処理で記録した最後のブロックラインのリスタートマーカの位置情報とが一致するか否かを判定して、一致すれば処理を終了し、一致しなければステップS429で次のストライプの処理に移り、上記したステップS421以下の処理を繰り返す。

【0034】このように、本実施形態によれば、リスタートマーカに基づいて、符号データを1ストライプ分ずつ復号して画像回転処理を行うので、バッファメモリに必要な記憶容量を削減することができる。例えば、ストライプの幅を256バイトにすれば、 $256 \times 4096 \times 3$ バイト、すなわち3Mバイトであり、従来方式（48Mバイト）に比べてバッファメモリの記憶容量を1/16にすることができる。

【0035】

【第2実施形態】以下、本発明にかかる第2実施形態の画像処理装置を説明する。なお、第2実施形態において、第1実施形態と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【0036】前述した第1実施形態においては、1ストライプごとにリスタートマーカを挿入する例を説明したが、本実施形態においては、図6に示すように、ブロックラインの末尾（または先頭）にリスタートマーカを挿入するものである。従って、符号化時の処理は、第1実施形態の処理に比べて若干簡略化される。

【0037】図7Aは本実施形態における符号化処理の手

順例を示すフローチャートで、前述した制御部521により制御・実行されるものであり、図5Aに示した処理手順と異なる部分だけ説明する。

【0038】図5AのステップS407では、16MCU分の符号化が終了したか、または、ブロックラインの符号化が終了したかを判定したが、図7AのステップS407aでは、ブロックラインの符号化が終了したか否かだけを判定する。

【0039】そして、ブロックラインの符号化が終了すると、ステップS408でリスタートマーカを挿入し、図5AのステップS409に対応する処理はなく、続いて、ステップS410で各ブロックラインごとのポインタにリスタートマーカの位置情報を格納した後、ステップS401へ戻る。

【0040】図7Bは本実施形態における画像回転処理の手順例を示すフローチャートで、前述した制御部521により制御・実行されるものである。

【0041】ステップS801で一つのブロックラインの1MCU分の符号データを復号し、ステップS802で復号済みの符号データの位置情報をブロックラインに対応するポインタにセットし、ステップS803で復号した画像データを出力バッファメモリ511へ格納し、ステップS804でブロックラインに対応する一組のレジスタに復号したブロックのY, U, VそれぞれのDC成分を格納し、ステップS805で次のブロックラインがあるか否かを判定し、あればステップS806で次のブロックラインの処理へ移り、ステップS801からS805の処理を繰り返す。なお、レジスタは制御部521のRAMなどに割当てて。

【0042】1MCU幅に相当するすべてのブロックラインの処理が終了すると、ステップS807で前述した画像回転処理を行い、ステップS808でリスタートマーカを検出したか否かを判定し、検出した場合は処理を終了し、そうでなければステップS801へ戻る。

【0043】つまり、本実施形態においては、1MCU幅の画像データを1ストライプとして処理するものである。そして、ブロックラインに対応する一組のレジスタに格納したY, U, VそれぞれのDC成分により、ブロックラインに対応するポインタに基づいて画像メモリ507から読出した符号データを復号する。そして、1MCUの復号処理ごとにブロックラインを切替えるため、ブロックラインを切替えるためにリスタートマーカを検出する必要はなく、すべての符号データが復号されたか否かを検出するためにリスタートマーカを用いる。

【0044】なお、第1実施形態においてポインタに格納するリスタートマーカの位置情報は、リスタートマーカの位置がバイト整列しているためバイト単位でよかったが、第2実施形態においてポインタに格納する位置情報は、それより細かいビット単位になる。といっても、増える情報は僅かに3ビットのみである。

【0045】このように、本実施形態によれば、1MCUに相当する幅のストライプ単位で復号し画像回転処理を行

うので、出力バッファメモリ511の記憶容量は、第1実施形態に比べて1/16にすることができる。

【0046】上述では、1MCUに相当する幅のストライプ単位で復号し画像回転処理を行う例を説明したが、MCU  $k(\geq 1)$ 個分に相当する幅のストライプ単位で復号し画像回転処理を行うようにしてもよい。このようにすると、出力バッファメモリ511の記憶容量は、1MCUに相当する幅のストライプ単位の処理に比べて $k$ 倍になるが、処理の切換えが減る分、処理スピードを速くすることができる。

【0047】

【第3実施形態】以下、本発明にかかる第3実施形態の画像処理装置を説明する。なお、第3実施形態において、第1実施形態と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【0048】次に、前述した第1または第2実施形態の画像回転処理を電子写真方式などのプリンタに適用する例を説明する。

【0049】一般に、A3記録紙に対応した電子写真方式のプリンタにおいては、A4記録紙を出力する際は、プリントスピードを速くするために、記録紙の向きをランドスケープにしてプリント出力することが多い。ところで、ホストコンピュータからプリンタに送られるてくるデータは、ほとんどの場合、記録紙の向きがポートレートであることを前提としている。つまり、プリンタが実際に出力する記録紙の向きと、ホストコンピュータからプリンタへデータを転送する際に前提とする記録紙の向きとが違っていることになる。そこで、画像の回転処理が必要になるわけである。

【0050】ホストコンピュータからプリンタへ転送されるデータは、PostScriptやPCLなどに代表されるいわゆるPDLコードと画像データに大別される。前者は、グラフィックコマンドや文字コード等から構成され、ビットマップデータを作る際に、所謂描画処理が必要になる。従って、描画処理を行う際に、座標系のとり方で記録紙の向きの違いを容易に吸収することができる。これに対して、後者は、ホストコンピュータからプリンタへ転送される時点で、既にビットマップデータになっているため、転送動作後に画像回転処理が必要になる。

【0051】図8は上記の二種類のデータをホストコンピュータから受取り、そのうち画像データをJPEG符号化・復号し、画像回転処理を施すプリンタの構成例を示すブロック図である。なお、図4と略同一の構成には、同一符号を付し、その詳細説明を省略する。

【0052】同図において、905はホストコンピュータから送られてきたPDLコードに従ってビットマップデータを描画する描画部、909はプリンタエンジン911へビットマップデータを出力するためのエンジンインタフェースである。

【0053】インタフェース901を介してホストコンピ

ュータから受取ったデータは、PDLコードと画像データを区別して入力バッファメモリ503に格納される。入力バッファメモリ503に格納された画像データは、例えば16ライン分のデータごとに符号化部505に読出され、前述したように圧縮された後、画像メモリ507に蓄えられる。画像メモリ507に格納された圧縮データは、復号部509により読出され、出力バッファメモリ511を用いて前述した復号および画像回転処理が施された後、プリンタエンジン911の印刷スピードに合わせて読出され、エンジンインタフェイス909を介して、プリンタエンジン911へ送られる。

【0054】一方、PDLコードの場合は、一頁分の画像に相当する量のPDLコードが入力バッファメモリ503に蓄積され、プリンタエンジン911へ先に出力すべきものから順に描画するために、実際の描画処理に先立ち、予めデータをソーティングされる。その際、記録紙の向きを合わせるために、必要ならば座標系を回転する。

【0055】一頁分に相当するPDLコードの受信が終了すると、描画部905によりPDLコードがビットマップデータに変換され、出力バッファメモリ511へ格納される。出力バッファメモリ511に格納されたビットマップデータは、プリンタエンジン911の印刷スピードに合わせて読出され、エンジンインタフェイス909を介して、プリンタエンジン911へ送られる。

#### 【0056】

【他の実施形態】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダー、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0057】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、

CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0058】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0059】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0060】

【発明の効果】以上説明したように、本発明によれば、画像データに画像回転処理を施す画像処理装置およびその方法において、画像データを格納するメモリの記憶容量を削減することができる。

#### 【図面の簡単な説明】

【図1】画像回転処理を説明するフローチャート、

【図2】図1に示す画像回転処理に、JPEG符号化を適用した例を示すフローチャート、

【図3】本発明にかかる一実施形態におけるリスタートマーカの挿入位置を説明するための図、

【図4】本実施形態の画像処理装置の構成例を示すブロック図、

【図5A】本実施形態における符号化処理の手順例を示すフローチャート、

【図5B】本実施形態における画像回転処理の手順例を示すフローチャート、

【図6】第2実施形態におけるリスタートマーカの挿入位置を説明するための図、

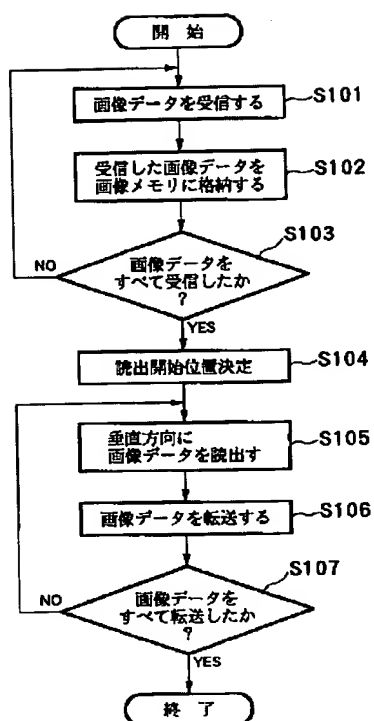
【図7A】第2本実施形態における符号化処理の手順例を示すフローチャート、

【図7B】第2実施形態における画像回転処理の手順例を示すフローチャート、

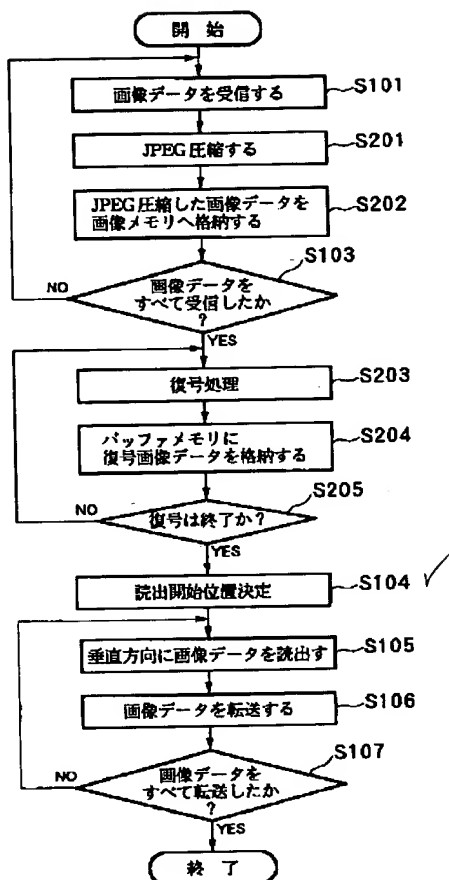
【図8】第3実施形態の画像回転処理を施すプリンタの構成例を示すブロック図である。



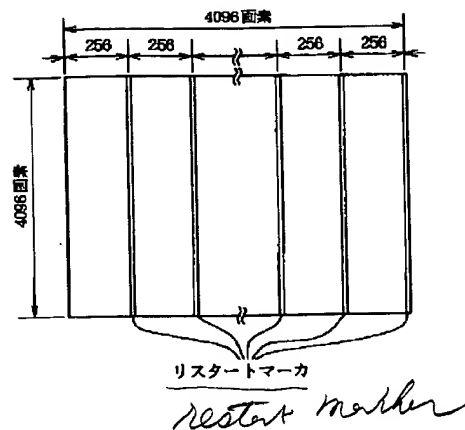
【図 1】



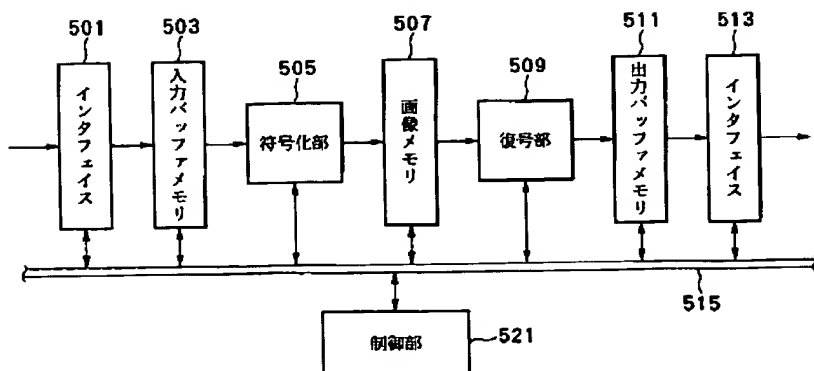
【図 2】



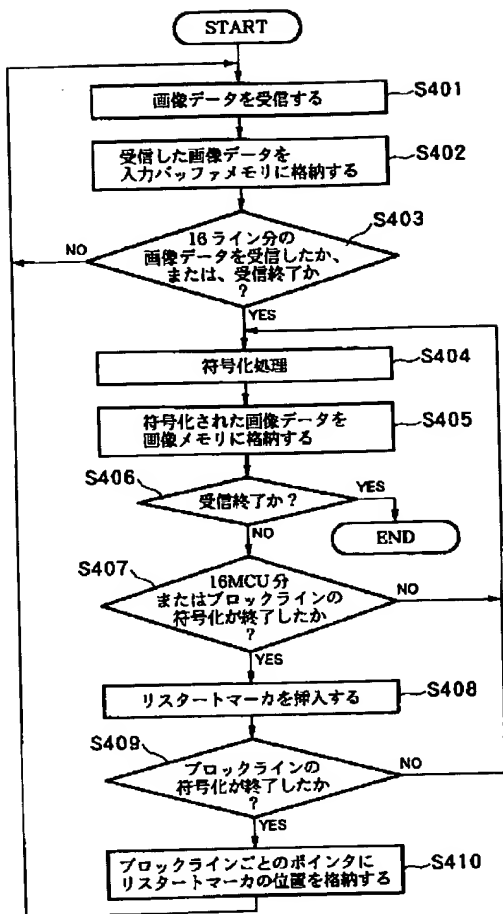
【図 3】



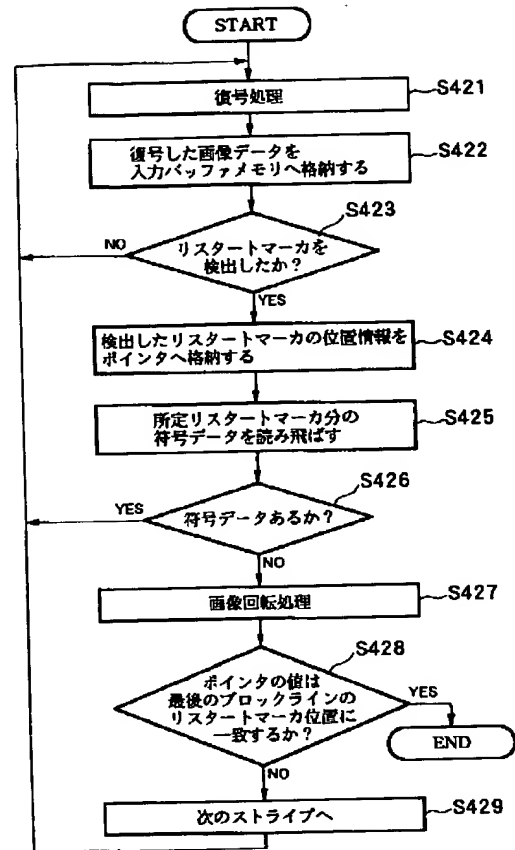
【図 4】



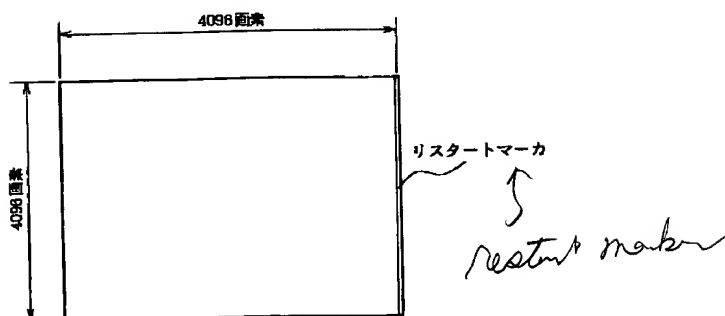
【図5A】



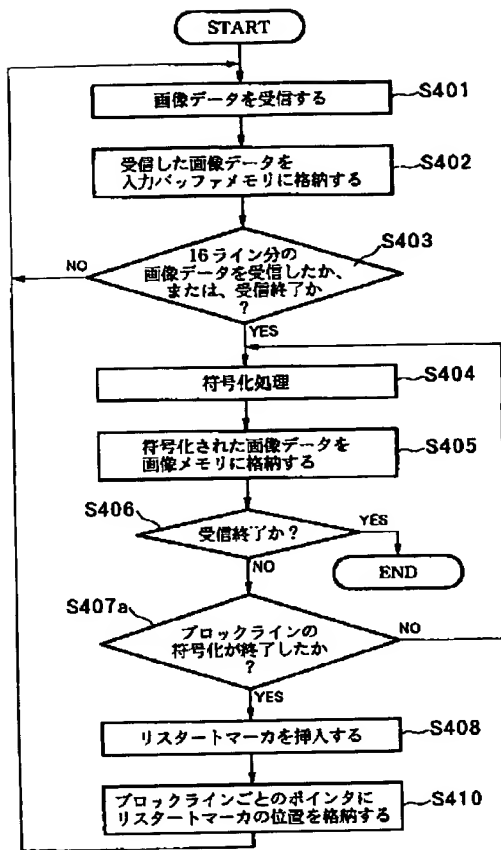
【図5B】



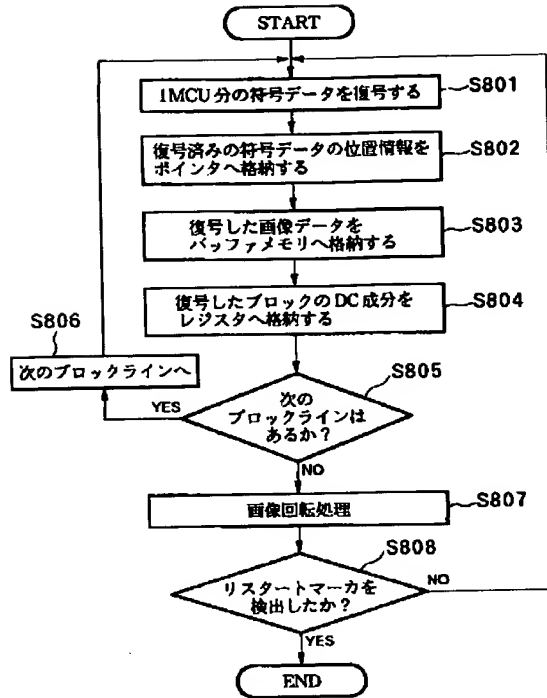
【図6】



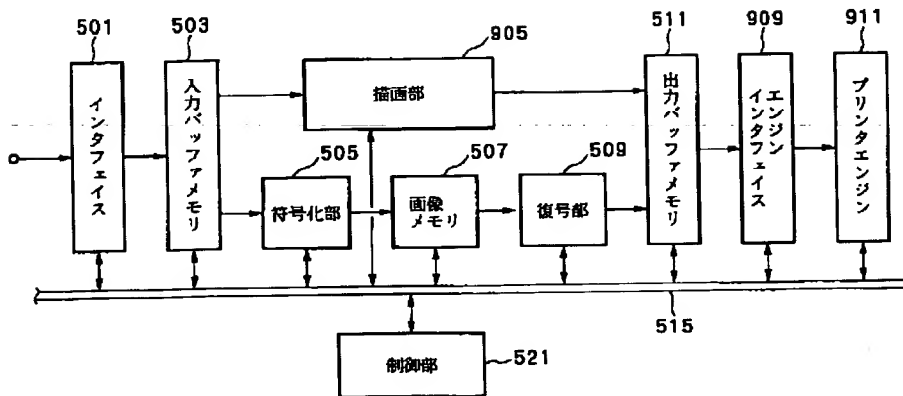
【図7A】



【図7B】



【図8】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**